

“A regular expression is a specific pattern used in computing that provides concise and flexible means to "match" (specify and recognize) strings of text, such as particular characters, words, or patterns of characters. Common abbreviations for "regular expression" include *regex* and *regexp*.”

Wikipedia

Character	Description	Example
. (dot)	Match any single character, except newline (\n)	<code>c.t</code> matches "cat", "cut" or "cot."
* (star)	Repeat the previous expression 0 or more times (greedy mode)	<code>12*3</code> matches "13", "123", "1223", "12223". It can be used together with . (dot) such as <code>m.*easier</code> to match "maketecheasier".
+ (plus)	Repeat the previous expression 1 or more times.	<code>12+3</code> matches "123","1223","12223"
? (question mark)	Makes the previous item optional.	<code>ma?ke</code> matches "make", "mke"
^ (caret)	Match from the beginning of the string	<code>^he</code> matches "hello", "hell", "help", "he is a boy"
\$ (dollar)	Match from the end of the string	<code>ed\$</code> matches "acted", "bed", "greed"
(. . .) (round bracket)	Grouping of characters or expression	<code>(ak)</code> matches "make", "take", "
{n} (curly bracket, where n >= 1)	Match the previous item exactly n times	<code>12{3}5</code> matches "12225"
{n,m} (n >=1, m > n)	Match the previous item between n and m times	<code>12{3,5}3</code> matches "12223", "122223", "1222223"
{n,} (n >=1)	Match the previous item at least n times	<code>12{2,}3</code> matches "1223", "12223", "122223"
[. . .] (square bracket)	match any single character in the bracket	<code>a[BC]d</code> matches "aBcd" or "abCd"
[^ . . .]	Match any character except for those that are defined in the bracket	<code>a[^b]c</code> matches "aec", "acc", "adc", but not "abc"
(pipe)	Match either the expression on the left or right of the pipe.	<code>col(o ou)r</code> matches "color", "colour"
- (hyphen)	Specify a range of characters to match. Used mostly with square bracket.	<code>[a-z]</code> matches all lowercase alphabet. <code>[A-Z]</code> matches all uppercase alphabet. <code>[0-9]</code> matches all the digit 0 to 9.
\ (backslash)	Escape a special	<code>a*c</code> matches "a*c".

Character	Description	Example
	character	
<code>\n, \r, \t, \d, \w, \s</code>	match a newline, return, tab, digit, word, whitespace character respectively	
<code>\D, \W, \S</code>	Negate version of <code>\d, \w, \s</code> . Match (not digit), (not word), (not whitespace) character respectively.	
<code>\b...\b</code>	Match a string at the boundary Can be used to match a word in a phrase.	<code>\bTech\b</code> matches the word "Tech" in the phrase "Make Tech Easier".
<code>\B...\B</code>	Match a string not within the boundary. Can be used to match a string within another string	<code>\BTech\B</code> matches "Tech" in "MakeTechEasier", but not in "Make Tech Easier".
<code>[\b]</code>	When used inside a square bracket, <code>\b</code> matches a backspace character	

Do you like this cheat sheet?

Visit MakeTechEasier.com for more useful tech tutorials, tips and tricks